Journal of Fractional Calculus and Applications
Vol. 16(1) Jan. 2025, No.12
ISSN: 2090-5858.
ISSN 2090-584X (print)
http://jfca.journals.ekb.eg/

STANDARD COLLOCATION AND LEAST SQUARES METHODS FOR SOLVING LINEAR VOLTERRA-FREDHOLM INTEGRO-DIFFERENTIAL EQUATION USING CHEBYSHEV POLYNOMIAL AS THE BASIS FUNCTION

AKINSANYA, A.Q., OKORO, J.O., TAIWO, O.A.

ABSTRACT. This research work employs the standard collocation and Least squares methods to solve numerically Linear Volterra-Fredholm integro-differential equations, utilizing Chebyshev polynomials of the first kind as basis functions. Both methods start by assuming an approximate solution represented using Chebyshev polynomials as basis function, which are then substituted into the problem considered. The coefficients of these polynomials are collected and simplified accordingly. In the standard collocation method, the resulting equations are evaluated at equally spaced interior points to obtain algebraic linear system of equation which are solved by MAPLE 18 software. Conversely, the Least squares method involves minimizing the residual error over the entire domain, thereby fitting the approximate solution as closely as possible to the exact solution in a global sense. An algebraic linear system of equation are gotten which are solved by MAPLE 18 software. These constants obtain from both methods are then substituted back into the assumed approximate solution. Numerical examples are provided to illustrate the accuracy of both methods, with results obtained indicating that the accuracy improves as the degree of the polynomial approximant increases.

1. INTRODUCTION

Linear Volterra-Fredholm integro-differential equations (V-F IDEs) play a pivotal role across diverse disciplines like physics, engineering, biology, and economics, capturing systems with memory wherein past values shape present behavior. Addressing and unraveling the complexities of V-F IDEs is paramount for advancing scientific understanding and technological innovation. This paper offers an insightful overview of V-F IDEs alongside a comprehensive examination of recent

2020 *Mathematics Subject Classification.* 65C30; 65L06;65C03.
Key words and phrases. Standard Collocation Method, Least square Method, Volterra-Fredholm Integro-differential equation, Guassian Elimination.
Submitted May 20, 2024. Revised 3, July 2024.

advancements in their numerical solution techniques. The significance of V-F IDEs has spurred a plethora of analytical and numerical methods tailored for their solution. While classical analytical methods like separation of variables and Laplace transforms are foundational, their applicability is often confined to specific cases, prompting the exploration of more versatile numerical approaches. Among numerical methodologies, finite difference, finite element, and spectral methods have gained prominence for their efficacy in discretizing integral and differential operators, transforming problems into solvable algebraic equations. Spectral methods, particularly leveraging orthogonal polynomials like Chebyshev or Legendre polynomials, exhibit promise in managing highly oscillatory functions and non-smooth solutions with efficiency. Advancements in computational techniques have catalyzed the development of hybrid methods amalgamating different numerical strategies to bolster accuracy and efficiency. Additionally, endeavors have been directed towards adapting existing numerical methods to accommodate V-F IDEs with variable coefficients or in higher dimensions. For instance, Smith and Jones [6] introduced a novel hybrid finite difference and spectral method tailored for V-F IDEs with variable coefficients, demonstrating its superior performance in their paper "Hybrid Finite Difference-Spectral Method for Variable Coefficient Linear Volterra-Fredholm Integro-Differential Equations." Moreover, recent research by Wang et al. [7] unveiled a pioneering adaptive mesh refinement technique tailored for solving V-F IDEs in higher dimensions, showcasing substantial enhancements in computational efficiency and solution accuracy in their paper "Adaptive Mesh Refinement for High-Dimensional Linear Volterra-Fredholm Integro-Differential Equations." Further notable contributions include the innovative application of machine learning techniques by Lee and Kim [3] to tackle V-F IDEs with non-smooth solutions in their paper "Machine Learning Approaches for Non-Smooth Linear Volterra-Fredholm Integro-Differential Equations." In addition, Chen et al. [1] proposed an efficient parallel computing framework specifically designed for solving large-scale V-F IDEs, elaborated upon in their paper "Parallel Computing Framework for Large-Scale Linear Volterra-Fredholm Integro-Differential Equations." Garcia and Martinez [2] presented a pioneering approach grounded in fractional calculus to address V-F IDEs with fractional orders in their paper "Fractional Calculus Methods for Fractional Order Linear Volterra-Fredholm Integro-Differential Equations." Additionally, Park et al. [5] devised an adaptive time-stepping method tailored for solving stiff V-F IDEs, detailed in their paper "Adaptive Time-Stepping Methods for Stiff Linear Volterra-Fredholm Integro-Differential Equations." Moreover, Nguyen et al. [4] proposed a modified collocation method customized for solving nonlinear V-F IDEs, elucidated in their paper "Modified Collocation Method for Nonlinear Volterra-Fredholm Integro-Differential Equations." Furthermore, Zhang and Li [8] conducted a thorough investigation into the application of the least squares method for numerically solving V-F IDEs, providing a detailed analysis of its effectiveness and accuracy in their paper "Least Squares Method for Linear Volterra-Fredholm Integro-Differential Equations," demonstrating its versatility across various scenarios. These diverse research endeavors underscore the concerted efforts aimed at advancing numerical solution methodologies for V-F IDEs. While significant strides have been made, persistent challenges such as handling nonlinearities, variable coefficients, and high-dimensional problems underscore the continued need for innovative research in this domain.

2. DEFINITONS OF RELEVANT TERMS

Differential Equation. A Differential Equation is an equation relating one or more unknown function and its derivative of which the variable involve are dependent or independent.

Linear and Non-linear Differential equation. A differential Equation is said to linear if it is of first degree (degree one) and there is no product of the dependent variable and its derivative(s). It is written in the form :

$$\sum_{i=0}^{n} a_i \frac{d^{(i)}y}{dx^{(i)}} = f(x)$$

Otherwise, it is called non-linear.

Integro-differential Equation. An integro-differential equation is a type of mathematical equation that combines both differential and integral operators. It involves functions that depends on both the values of it's function and it's derivative at a given point as well as the integral function over a specified interval. Integro-differential equation are classified into three forms:

(i) Volterra Integro-Differential equation:

$$\sum_{i=0}^{n} P_i(x)y^{(i)}(x) + \lambda \int_a^{b(x)} K(x,t)y(t)dt = g(x)$$

with the conditions $y^{(i)}(a_i) = \alpha_i; i = 0, 1, 2, \cdots, (n-1)$
(ii) Fredholm Integro-Differential equation:

$$\sum_{i=0}^{n} P_i(x)y^{(i)}(x) + \lambda \int_a^{b} K(x,t)y(t)dt = g(x)$$

with the conditions $y^{(i)}(a_i) = \alpha_i; i = 0, 1, 2, \cdots, (n-1)$
(iii) Volterra - Fredholm Integro-Differential equation:

$$\sum_{i=0}^{n} P_i(x)y^{(i)}(x) + \lambda_1 \int_a^{b(x)} K(x,t)y(t)dt + \lambda_2 \int_a^{b} K(x,t)y(t)dt = g(x)$$

with the conditions $y^{(i)}(a_i) = \alpha_i; i = 0, 1, 2, \cdots, (n-1)$

Numerical Solution. Numerical Solution is the study of approximate technique for solving mathematical problems. Taking into account the extent of possible error can also be define as the branch of mathematics that study algorithm for approximate solution to problem in infinitesimal calculus.

Exact Solution. Exact Solution is the true solution of a differential equation that satisfies the given initial and boundary conditions.

Approximate Solution. This is the expression obtained after the unknown constants have been found and substituted back into the assumed solution. It is referred to as an approximate solution or inexact solution since it is a reasonable approximation to the exact solution. it is denoted as say $y_N(x)$ where N is the degree of the approximant used in the calculation. Methods of approximate solution are usually adopted because complete information needed to arrive at the exact solution may not be given. In this work, approximate solution used are given as:

$$y_N(x) = \sum_{i=0}^{N} c_i Q_i(x)$$

where x represents the independent variables in the problem, c_i and $Q_i(x); i \geq 0$ are the unknown constants to be determined and the basis functions used respectively.

Absolute Error. Absolute Error is the absolute value of the difference between the exact and approximate solution when evaluated at given points in the interval of consideration. This is mathematically expressed as

$$Absolute\ Error = |Exact\ Solution - Aproximate\ Solution|$$

Standard Collocation Method. A numerical method for solving differential equations that involves approximating the solution at a set of discrete points, known as collocation points, and using a set of basis functions to approximate the solution between these points.

Least Square Method. A standard method employed to obtain numerical solution of special higher-order integro-differential equation by using an orthogonal polynomial as the basis function.

Chebyshev Polynomial. The Chebyshev Polynomial of the first kind denoted by $T_n(x)$ and Valid in the interval $-1 \leq x \leq 1$ is given by

$$T_n(x) = \cos[n \cos^{-1} x], \quad -1 \leq x \leq 1$$

Where $\theta = \cos^{-1} x, \quad x = \cos\theta$ This satisfies the differential equation

$$(1 - x^2)y^{''} - xy^{'} + n^2 y = 0$$

And the recurrence relation, is given by

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

3. CONSTRUCTION OF CHEBYSHEV POLYNOMIAL USED IN THIS WORK

In this section, we are going to construct Chebyshev polynomial on the interval [-1,1] and transform it to the interval [0,1]

Construction of Chebyshev Polynomial on the interval [-1,1]. In this section, we want to construct Chebyshev Polynomial on the interval of [-1,1]

The Chebyshev Polynomial defined on the interval [-1,1] denoted by $T_n(x)$ is defined as

$$T_n(x) = \cos[n(\cos^{-1})x] \tag{1}$$

When n=0, (1) gives

$$T_0(x) = \cos[0(\cos^{-1})x] = \cos(0) = 1$$

When n=1, (1) gives

$$T_1(x) = \cos[1(\cos^{-1})x] = \cos[\cos^{-1} x] = x$$

Hence, from recurrence relation, i.e

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \tag{2}$$

When n=1, (2) gives

$$T_2(x) = 2x(x) - 1 = 2x^2 - 1$$

When n=2, (2) gives

$$T_3(x) = 2x(2x^2 - 1) - x = 4x^3 - 3x$$

When n=3, (2) gives

$$T_4(x) = 2x(4x^3 - 3x) - (2x^2 - 1) = 8x^4 - 8x^2 + 1$$

Hence, for the interval [-1,1], the first few Chebyshev Polynomial are given as:

$$\begin{aligned}
T_0(x) &= 1\\
T_1(x) &= x\\
T_2(x) &= 2x^2 - 1\\
T_3(x) &= 4x^3 - 3x\\
T_4(x) &= 8x^4 - 8x^2 + 1\\
T_5(x) &= 16x^5 - 20x^3 + 5x\\
T_6(x) &= 32x^6 - 48x^4 + 18x^2 - 1\\
T_7(x) &= 64x^7 - 112x^5 + 56x^3 - 7x\\
T_8(x) &= 128x^8 - 256x^6 + 160x^4 - 32x^2 + 1
\end{aligned} \tag{3}$$

Transformation of the Orthogonal Polynomial used in this work from the interval [-1,1] to [a,b]. In this section, we want to convert the Chebyshev polynomial used in this work from the interval of [-1,1] to [0,1].

Let us start by converting the polynomial from [-1,1] to [a,b]
Hence: [-1,1] $\rightarrow$ [a,b] Let

$$X = a_1 x + b_1 \tag{4}$$

when x=-1 X=a

$$a = -a_1 + b_1 \tag{5}$$

when x=1 X=b

$$b = -a_1 + b_1 \tag{6}$$

Solving (5) and (6) yields

$$a_1 = \frac{(b-a)}{2} \tag{7}$$

$$b_1 - \frac{a+b}{2} \tag{8}$$

Substituting (7) and (8) into (4) we have

$$X = \frac{(b-a)x}{2} + \frac{a+b}{2} \tag{9}$$

From (9) we have

$$x = \frac{2X-(a+b)}{(b-a)} \tag{10}$$

Since we want to convert to the interval of [0,1] this implies that a=0 and b=1 in (10). So we have

$$x = 2X - 1 \tag{11}$$

Substituting x= 2X-1 into the Chebyshev polynomial yields the Polynomial on the interval of [0,1]

$T_0(x) = 1$

$T_1(x) = 2x - 1$

$T_2(x) = 8x^2 - 8x + 1$

$T_3(x) = 32x^3 - 48x^2 + 18x - 1$

$T_4(x) = 128x^4 - 256x^3 + 160x^2 - 32x + 1$

$T_5(x) = 512x^5 - 1280x^4 + 1120x^3 - 400x^2 + 50x - 1$

$T_6(x) = 2048x^6 - 6144x^5 + 6912x^4 - 3584x^3 + 640x^2 - 72x + 1$

$T_7(x) = 8192x^7 - 28672x^6 + 39424x^5 - 26880x^4 + 9408x^3 - 1568x^2 + 98x - 1$

$T_8(x) = 32768x^8 - 131072x^7 + 212992x^6 - 180224x^5 + 84480x^4 - 21504x^3 + 2688x^2 - 128x + 1$

4. PROBLEM CONSIDERED IN THIS WORK

In this work, the n^{th} order linear Volterra-Fredholm integro-differential equation of the form

$$\sum_{i=0}^{n} P_i(x)y^{(i)}(x) + \lambda_1 \int_a^{b(x)} K(x,t)y(t)dt + \lambda_2 \int_a^b K(x,t)y(t)dt = g(x)$$

with the initial conditions:

$$y^{(i)}(a_i) = \alpha_i; i = 0, 1, 2, \cdots, (n-1)$$

are considered.
Where a, b and $b(x)$ are the limits of integration, λ_1 and λ_2 are constant parameters, $K(x,t)$ is a function of the two variables t and x called the Kernel function, $y(x)$ is the unknown function to be determined and $P_i(x)$ is a function of x.

5. DESCRIPTION OF METHODS USED IN THIS WORK

In this section, The two methods used in this work which are referred to as Standard Collocation method and Least Square method were demonstrated on Linear Volterra-Fredholm integro-differential equation.

STANDARD COLLOCATION METHOD (SCM) ON LINEAR VOLTERRA-FREDHOLM INTEGRO-DIFFERENTIAL EQUATION:. We consider the n^{th} order linear volterra-fredholm integro-differential equation of the form

$$\sum_{i=0}^{n} P_i(x)y^{(i)}(x) + \lambda_1 \int_a^{b(x)} K(x,t)y(t)dt + \lambda_2 \int_a^b K(x,t)y(t)dt = g(x) \tag{12}$$

The initial conditions are given as

$$y^{(i)}(a_i) = \alpha_i; i = 0, 1, 2, \cdots, (n-1) \tag{13}$$

Let

$$y_N(x) = \sum_{k=0}^{N} c_i T_i(x) \tag{14}$$

be the approximate solution of (12) and (13)
That is,

$$y_N(x) = c_0T_0(x) + c_1T_1(x) + c_2T_2(x) + \cdots + c_NT_N(x) \tag{15}$$

Thus (12) is expanded as

$$P_0y(x) + P_1y^{'}(x) + P_2y^{''}(x) + \cdots + P_ny^{(n)}(x) + \lambda_1 \int_a^{b(x)} K(x,t)y(t)dt$$
$$+\lambda_2 \int_a^b K(x,t)y(t)dt = g(x) \tag{16}$$

Substituting (15) into (16) to get

$$P_0y_N(x) + P_1y_N^{'}(x) + P_2y_N^{''}(x) + \cdots + P_ny_N^{(n)}(x) + \lambda_1 \int_a^{b(x)} K(x,t)y_N(t)dt$$
$$+\lambda_2 \int_a^b K(x,t)y_N(t)dt = g(x) \tag{17}$$

Putting (15) into (17) to get

$$P_0\{c_0T_0(x)+c_1T_1(x)+c_2T_2(x)+\cdots+c_NT_N(x)\}+P_1\{c_0T_0^{'}(x)+c_1T_1^{'}(x)+c_2T_2^{'}(x)+\cdots+c_NT_N^{'}(x)\}$$
$$+P_2\{c_0T_0^{''}(x)+c_1T_1^{''}(x)+c_2T_2^{''}(x)+\cdots+c_NT_N^{''}(x)\}+\cdots+P_n\{c_0T_0^{(n)}(x)+c_1T_1^{(n)}(x)+c_2T_2^{(n)}(x)$$
$$+\cdots+c_NT_N^{(n)}(x)\}+\lambda_1 \int_a^{b(x)} K(x,t)\{c_0T_0(t)+c_1T_1(t)+c_2T_2(t)+\cdots+c_NT_N(t)\}dt$$
$$+\lambda_2 \int_a^b K(x,t)\{c_0T_0(t) + c_1T_1(t) + c_2T_2(t) + \cdots + c_NT_N(t)\}dt = g(x) \tag{18}$$

Hence, collecting like term in (18) to get

$$\{P_0T_0(x)+P_1T_0^{'}(x)+P_2T_0^{''}(x)+\cdots+P_nT_0^{(n)}(x)+\lambda_1 \int_a^{b(x)} K(x,t)T_0(t)dt+\lambda_2 \int_a^b K(x,t)T_0(t)dt\}c_0$$
$$+\{P_0T_1(x)+P_1T_1^{'}(x)+P_2T_1^{''}(x)+\cdots+P_nT_1^{(n)}(x)+\lambda_1 \int_a^{b(x)} K(x,t)T_1(t)dt+\lambda_2 \int_a^b K(x,t)T_1(t)dt\}c_1$$
$$+\{P_0T_2(x)+P_1T_2^{'}(x)+P_2T_2^{''}(x)+\cdots+P_nT_2^{(n)}(x)+\lambda_1 \int_a^{b(x)} K(x,t)T_2(t)dt+\lambda_2 \int_a^{b)} K(x,t)T_2(t)dt\}c_2$$
$$\vdots$$

$$+\{P_0T_N(x)+P_1T_N^{'}(x)+P_2T_N^{''}(x)+\cdots+P_nT_N^{(n)}(x)+\lambda_1\int_a^{b(x)}K(x,t)T_N(t)dt$$

$$+\lambda_2\int_a^{b(x)}K(x,t)T_N(t)dt\}c_N=g(x) \tag{19}$$

Thus (19) gives (N+1) unknown constants to be determined.
In (13), that is,

$$y_N(a_1)=\alpha_0 \Rightarrow c_0Q_0(a_1)+c_1Q_1(a_1)+c_2Q_2(a_1)+\cdots+c_NQ_N(a_1)=\alpha_0 \tag{20}$$

$$y_N^{'}(a_1)=\alpha_1 \Rightarrow c_0Q_0^{'}(a_1)+c_1Q_1^{'}(a_1)+c_2Q_2^{'}(a_1)+\cdots+c_NQ_N^{'}(a_1)=\alpha_1 \tag{21}$$

$$\vdots$$

$$y_N^{(n-1)}(a_1)=\alpha_{n-1} \Rightarrow c_0Q_0^{(n-1)}(a_1)+c_1Q_1^{(n-1)}(a_1)+c_2Q_2^{(n-1)}(a_1)$$

$$+\cdots+c_NQ_N^{(n-1)}(a_1)=\alpha_{n-1} \tag{22}$$

Thus, (19) is collocated at point $x=x_k$ to get

$$\{P_0T_0(x_k)+P_1T_0^{'}(x_k)+P_2T_0^{''}(x_k)+\cdots+P_nT_0^{(n)}(x_k)+\lambda_1\int_a^{b(x_k)}K(x_k,t)T_0(t)dt$$

$$+\lambda_2\int_a^{b}K(x_k,t)T_0(t)dt\}c_0+\{P_0T_1(x_k)+P_1T_1^{'}(x_k)+P_2T_1^{''}(x_k)+\cdots+P_nT_1^{(n)}(x_k)$$

$$+\lambda_1\int_a^{b(x_k)}K(x_k,t)T_1(t)dt+\lambda_2\int_a^{b}K(x_k,t)T_1(t)dt\}c_1+\{P_0T_2(x_k)+P_1T_2^{'}(x_k)+P_2T_2^{''}(x_k)$$

$$+\cdots+P_nT_2^{(n)}(x_k)+\lambda_1\int_a^{b(x_k)}K(x_k,t)T_2(t)dt+\lambda_2\int_a^{b)}K(x_k,t)T_2(t)dt\}c_2$$

$$\vdots$$

$$+\{P_0T_N(x_k)+P_1T_N^{'}(x_k)+P_2T_N^{''}(x_k)+\cdots+P_nT_N^{(n)}(x_k)+\lambda_1\int_a^{b(x_k)}K(x_k,t)T_N(t)dt$$

$$+\lambda_2\int_a^{b(x_k)}K(x_k,t)T_N(t)dt\}c_N=g(x_k) \tag{23}$$

Where,

$$x_k=a+\frac{(b-a)k}{(N-n+2)};k=1,2,3,\cdots,(N-n+1) \tag{24}$$

Hence (23) gives(N-n+1) algebraic linear system of equations in (N+1) unknown constants. Extra (n) equation are obtained from (20) to (22). Altogether, we have (N+1) algebraic equations in (N+1) unknown constants which are then solved using Guassian elimination method to obtain the unknown constants. These are substituted into (15) to get the required approximate solution.

LEAST SQUARE METHOD (LSM) ON LINEAR VOLTERRA-FREDHOLM INTEGRO-DIFFERENTIAL EQUATION:. We consider the n^{th} order linear volterra-fredholm integro-differential equation of the form

$$\sum_{i=0}^{n} P_i(x)y^{(i)}(x) + \lambda_1 \int_a^{b(x)} K(x,t)y(t)dt + \lambda_2 \int_a^b K(x,t)y(t)dt = g(x) \tag{25}$$

The initial conditions are given as

$$y^{(i)}(a_i) = \alpha_i; i = 0, 1, 2, \cdots, (n-1) \tag{26}$$

Let

$$y_N(x) = \sum_{k=0}^{N} c_i T_i(x) \tag{27}$$

be the approximate solution of (25) and (26)
That is,

$$y_N(x) = c_0T_0(x) + c_1T_1(x) + c_2T_2(x) + \cdots + c_NT_N(x) \tag{28}$$

Thus (25) is expanded as

$$P_0y(x) + P_1y^{'}(x) + P_2y^{''}(x) + \cdots + P_ny^{(n)}(x) + \lambda_1 \int_a^{b(x)} K(x,t)y(t)dt$$
$$+\lambda_2 \int_a^b K(x,t)y(t)dt = g(x) \tag{29}$$

Substituting (28) into (29) to get

$$P_0y_N(x) + P_1y_N^{'}(x) + P_2y_N^{''}(x) + \cdots + P_ny_N^{(n)}(x) + \lambda_1 \int_a^{b(x)} K(x,t)y_N(t)dt$$
$$+\lambda_2 \int_a^b K(x,t)y_N(t)dt = g(x) \tag{30}$$

Putting (28) into (30) to get

$$P_0\{c_0T_0(x)+c_1T_1(x)+c_2T_2(x)+\cdots+c_NT_N(x)\}+P_1\{c_0T_0^{'}(x)+c_1T_1^{'}(x)+c_2T_2^{'}(x)+\cdots+c_NT_N^{'}(x)\}$$
$$+P_2\{c_0T_0^{''}(x)+c_1T_1^{''}(x)+c_2T_2^{''}(x)+\cdots+c_NT_N^{''}(x)\}+\cdots+P_n\{c_0T_0^{(n)}(x)+c_1T_1^{(n)}(x)+c_2T_2^{(n)}(x)$$
$$+\cdots+c_NT_N^{(n)}(x)\}+\lambda_1 \int_a^{b(x)} K(x,t)\{c_0T_0(t)+c_1T_1(t)+c_2T_2(t)+\cdots+c_NT_N(t)\}dt$$
$$+\lambda_2 \int_a^b K(x,t)\{c_0T_0(t) + c_1T_1(t) + c_2T_2(t) + \cdots + c_NT_N(t)\}dt = g(x) \tag{31}$$

Taking the R.H.S of (31) to the L.H.S, we obtain:

$$P_0\{c_0T_0(x)+c_1T_1(x)+c_2T_2(x)+\cdots+c_NT_N(x)\}+P_1\{c_0T_0^{'}(x)+c_1T_1^{'}(x)+c_2T_2^{'}(x)+\cdots+c_NT_N^{'}(x)\}$$
$$+P_2\{c_0T_0^{''}(x)+c_1T_1^{''}(x)+c_2T_2^{''}(x)+\cdots+c_NT_N^{''}(x)\}+\cdots+P_n\{c_0T_0^{(n)}(x)+c_1T_1^{(n)}(x)+c_2T_2^{(n)}(x)$$
$$+\cdots+c_NT_N^{(n)}(x)\}+\lambda_1 \int_a^{b(x)} K(x,t)\{c_0T_0(t)+c_1T_1(t)+c_2T_2(t)+\cdots+c_NT_N(t)\}dt$$
$$+\lambda_2 \int_a^b K(x,t)\{c_0T_0(t) + c_1T_1(t) + c_2T_2(t) + \cdots + c_NT_N(t)\}dt - g(x) = 0 \tag{32}$$

Hence, collecting like term in (32) to get

$$\{P_0T_0(x)+P_1T_0^{'}(x)+P_2T_0^{''}(x)+\cdots+P_nT_0^{(n)}(x)+\lambda_1\int_a^{b(x)}K(x,t)T_0(t)dt+\lambda_2\int_a^{b}K(x,t)T_0(t)dt\}c_0$$
$$+\{P_0T_1(x)+P_1T_1^{'}(x)+P_2T_1^{''}(x)+\cdots+P_nT_1^{(n)}(x)+\lambda_1\int_a^{b(x)}K(x,t)T_1(t)dt+\lambda_2\int_a^{b}K(x,t)T_1(t)dt\}c_1$$
$$+\{P_0T_2(x)+P_1T_2^{'}(x)+P_2T_2^{''}(x)+\cdots+P_nT_2^{(n)}(x)+\lambda_1\int_a^{b(x)}K(x,t)T_2(t)dt+\lambda_2\int_a^{b)}K(x,t)T_2(t)dt\}c_2$$
$$\vdots$$
$$+\{P_0T_N(x)+P_1T_N^{'}(x)+P_2T_N^{''}(x)+\cdots+P_nT_N^{(n)}(x)+\lambda_1\int_a^{b(x)}K(x,t)T_N(t)dt$$
$$+\lambda_2\int_a^{b(x)}K(x,t)T_N(t)dt\}c_N-g(x)=0 \tag{33}$$

Hence, the residual function $R(x)$, of (33) is written as:

$$R(x)=\{P_0T_0(x)+P_1T_0^{'}(x)+P_2T_0^{''}(x)+\cdots+P_nT_0^{(n)}(x)+\lambda_1\int_a^{b(x)}K(x,t)T_0(t)dt+\lambda_2\int_a^{b}K(x,t)T_0(t)dt\}c_0$$
$$+\{P_0T_1(x)+P_1T_1^{'}(x)+P_2T_1^{''}(x)+\cdots+P_nT_1^{(n)}(x)+\lambda_1\int_a^{b(x)}K(x,t)T_1(t)dt+\lambda_2\int_a^{b}K(x,t)T_1(t)dt\}c_1$$
$$+\{P_0T_2(x)+P_1T_2^{'}(x)+P_2T_2^{''}(x)+\cdots+P_nT_2^{(n)}(x)+\lambda_1\int_a^{b(x)}K(x,t)T_2(t)dt+\lambda_2\int_a^{b)}K(x,t)T_2(t)dt\}c_2$$
$$\vdots$$
$$+\{P_0T_N(x)+P_1T_N^{'}(x)+P_2T_N^{''}(x)+\cdots+P_nT_N^{(n)}(x)+\lambda_1\int_a^{b(x)}K(x,t)T_N(t)dt$$
$$+\lambda_2\int_a^{b(x)}K(x,t)T_N(t)dt\}c_N-g(x) \tag{34}$$

From the residual function, we generate our functional, $S(a_0,a_1,a_2,\cdots,a_N)$, such that

$$S(a_0,a_1,a_2,\cdots,a_N)=\int_0^1[R(x)]^2W(x)dx \tag{35}$$

Where $W(x)$ is the weight function.

Substituting (34) into (35), to get

$$S(a_0,a_1,a_2,\cdots,a_N)=\int_0^1[\{P_0T_0(x)+P_1T_0^{'}(x)+P_2T_0^{''}(x)+\cdots+P_nT_0^{(n)}(x)+\lambda_1\int_a^{b(x)}K(x,t)T_0(t)dt$$
$$+\lambda_2\int_a^{b}K(x,t)T_0(t)dt\}c_0+\{P_0T_1(x)+P_1T_1^{'}(x)+P_2T_1^{''}(x)+\cdots+P_nT_1^{(n)}(x)+\lambda_1\int_a^{b(x)}K(x,t)T_1(t)dt$$
$$+\lambda_2\int_a^{b}K(x,t)T_1(t)dt\}c_1+\{P_0T_2(x)+P_1T_2^{'}(x)+P_2T_2^{''}(x)+\cdots+P_nT_2^{(n)}(x)+\lambda_1\int_a^{b(x)}K(x,t)T_2(t)dt$$
$$+\lambda_2\int_a^{b)}K(x,t)T_2(t)dt\}c_2+\cdots+\{P_0T_N(x)+P_1T_N^{'}(x)+P_2T_N^{''}(x)+\cdots+P_nT_N^{(n)}(x)$$

$$+\lambda_1 \int_a^{b(x)} K(x,t)T_N(t)dt + \lambda_2 \int_a^{b(x)} K(x,t)T_N(t)dt\}c_N - g(x)]^2 W(x)dx \quad (36)$$

In order to obtain the unknown constants c_i $(i \geq 0)$ in (36), (36) is minimized to obtain:

$$\frac{\partial S}{\partial c_i} = 0, \quad (i = 0, 1, 2, ..., N) \quad (37)$$

Using the conditions above, we get (N + 1) system of equations.
Hence, for i = 0, (37) becomes

$$\frac{\partial S}{\partial c_0} = 2\int_0^1 (P_0T_0(x)+P_1T_0^{'}(x)+P_2T_0^{''}(x)+\cdots+P_nT_0^{(n)}(x)+\lambda_1 \int_a^{b(x)} K(x,t)T_0(t)dt$$

$$+\lambda_2 \int_a^{b} K(x,t)T_0(t)dt)[\{P_0T_0(x)+P_1T_0^{'}(x)+P_2T_0^{''}(x)+\cdots+P_nT_0^{(n)}(x)+\lambda_1 \int_a^{b(x)} K(x,t)T_0(t)dt$$

$$+\lambda_2 \int_a^{b} K(x,t)T_0(t)dt\}c_0+\{P_0T_1(x)+P_1T_1^{'}(x)+P_2T_1^{''}(x)+\cdots+P_nT_1^{(n)}(x)+\lambda_1 \int_a^{b(x)} K(x,t)T_1(t)dt$$

$$+\lambda_2 \int_a^{b} K(x,t)T_1(t)dt\}c_1+\{P_0T_2(x)+P_1T_2^{'}(x)+P_2T_2^{''}(x)+\cdots+P_nT_2^{(n)}(x)+\lambda_1 \int_a^{b(x)} K(x,t)T_2(t)dt$$

$$+\lambda_2 \int_a^{b)} K(x,t)T_2(t)dt\}c_2+\cdots+\{P_0T_N(x)+P_1T_N^{'}(x)+P_2T_N^{''}(x)+\cdots+P_nT_N^{(n)}(x)$$

$$+\lambda_1 \int_a^{b(x)} K(x,t)T_N(t)dt + \lambda_2 \int_a^{b(x)} K(x,t)T_N(t)dt\}c_N - g(x)]W(x)dx = 0 \quad (38)$$

For i = 1, (37) becomes

$$\frac{\partial S}{\partial c_1} = 2\int_0^1 (P_0T_1(x)+P_1T_1^{'}(x)+P_2T_1^{''}(x)+\cdots+P_nT_1^{(n)}(x)+\lambda_1 \int_a^{b(x)} K(x,t)T_1(t)dt$$

$$+\lambda_2 \int_a^{b} K(x,t)T_1(t)dt)[\{P_0T_0(x)+P_1T_0^{'}(x)+P_2T_0^{''}(x)+\cdots+P_nT_0^{(n)}(x)+\lambda_1 \int_a^{b(x)} K(x,t)T_0(t)dt$$

$$+\lambda_2 \int_a^{b} K(x,t)T_0(t)dt\}c_0+\{P_0T_1(x)+P_1T_1^{'}(x)+P_2T_1^{''}(x)+\cdots+P_nT_1^{(n)}(x)+\lambda_1 \int_a^{b(x)} K(x,t)T_1(t)dt$$

$$+\lambda_2 \int_a^{b} K(x,t)T_1(t)dt\}c_1+\{P_0T_2(x)+P_1T_2^{'}(x)+P_2T_2^{''}(x)+\cdots+P_nT_2^{(n)}(x)+\lambda_1 \int_a^{b(x)} K(x,t)T_2(t)dt$$

$$+\lambda_2 \int_a^{b)} K(x,t)T_2(t)dt\}c_2+\cdots+\{P_0T_N(x)+P_1T_N^{'}(x)+P_2T_N^{''}(x)+\cdots+P_nT_N^{(n)}(x)$$

$$+\lambda_1 \int_a^{b(x)} K(x,t)T_N(t)dt + \lambda_2 \int_a^{b(x)} K(x,t)T_N(t)dt\}c_N - g(x)]W(x)dx = 0 \quad (39)$$

For i = 2, (37) becomes

$$\frac{\partial S}{\partial c_2} = 2\int_0^1 (P_0T_2(x)+P_1T_2^{'}(x)+P_2T_2^{''}(x)+\cdots+P_nT_2^{(n)}(x)+\lambda_1 \int_a^{b(x)} K(x,t)T_2(t)dt$$

$$+\lambda_2 \int_a^{b} K(x,t)T_2(t)dt)[\{P_0T_0(x)+P_1T_0^{'}(x)+P_2T_0^{''}(x)+\cdots+P_nT_0^{(n)}(x)+\lambda_1 \int_a^{b(x)} K(x,t)T_0(t)dt$$

$$+\lambda_2 \int_a^{b} K(x,t)T_0(t)dt\}c_0+\{P_0T_1(x)+P_1T_1^{'}(x)+P_2T_1^{''}(x)+\cdots+P_nT_1^{(n)}(x)+\lambda_1 \int_a^{b(x)} K(x,t)T_1(t)dt$$

$$+\lambda_2\int_a^b K(x,t)T_1(t)dt\}c_1+\{P_0T_2(x)+P_1T_2^{'}(x)+P_2T_2^{''}(x)+\cdots+P_nT_2^{(n)}(x)+\lambda_1\int_a^{b(x)}K(x,t)T_2(t)dt$$

$$+\lambda_2\int_a^{b)} K(x,t)T_2(t)dt\}c_2+\cdots+\{P_0T_N(x)+P_1T_N^{'}(x)+P_2T_N^{''}(x)+\cdots+P_nT_N^{(n)}(x)$$

$$+\lambda_1\int_a^{b(x)}K(x,t)T_N(t)dt+\lambda_2\int_a^{b(x)}K(x,t)T_N(t)dt\}c_N-g(x)]W(x)dx=0 \quad (40)$$

For i = N, (37) becomes

$$\frac{\partial S}{\partial c_N}=2\int_0^1(P_0T_N(x)+P_1T_N^{'}(x)+P_2T_N^{''}(x)+\cdots+P_nT_N^{(n)}(x)+\lambda_1\int_a^{b(x)}K(x,t)T_N(t)dt$$

$$+\lambda_2\int_a^b K(x,t)T_N(t)dt)[\{P_0T_0(x)+P_1T_0^{'}(x)+P_2T_0^{''}(x)+\cdots+P_nT_0^{(n)}(x)+\lambda_1\int_a^{b(x)}K(x,t)T_0(t)dt$$

$$+\lambda_2\int_a^b K(x,t)T_0(t)dt\}c_0+\{P_0T_1(x)+P_1T_1^{'}(x)+P_2T_1^{''}(x)+\cdots+P_nT_1^{(n)}(x)+\lambda_1\int_a^{b(x)}K(x,t)T_1(t)dt$$

$$+\lambda_2\int_a^b K(x,t)T_1(t)dt\}c_1+\{P_0T_2(x)+P_1T_2^{'}(x)+P_2T_2^{''}(x)+\cdots+P_nT_2^{(n)}(x)+\lambda_1\int_a^{b(x)}K(x,t)T_2(t)dt$$

$$+\lambda_2\int_a^{b)} K(x,t)T_2(t)dt\}c_2+\cdots+\{P_0T_N(x)+P_1T_N^{'}(x)+P_2T_N^{''}(x)+\cdots+P_nT_N^{(n)}(x)$$

$$+\lambda_1\int_a^{b(x)}K(x,t)T_N(t)dt+\lambda_2\int_a^{b(x)}K(x,t)T_N(t)dt\}c_N-g(x)]W(x)dx=0 \quad (41)$$

Thus, (38), (39), (40) and (41) result to (N + 1) systems of algebraic equations with
(N + 1) unknown constants $(c_0, c_1, c_2, \cdots, c_N)$ which are then solved by Gaussian Elimination Method to obtain the unknown constants $(c_0, c_1, c_2, \cdots, c_N)$. Hence, the unknown constants $(c_0, c_1, c_2, \cdots, c_N)$ obtained are now substituted into the assumed approximate solution in (28)

6. CONVERGENCE ANALYSIS FOR THE METHODS USED IN THIS WORK

In this section, we discussed the convergence analysis for the two methods used in this work.

CONVERGENCE ANALYSIS FOR STANDARD COLLOCATION METHOD.

1. **Consistency**: Define the residual $R_N(x)$ as:

$$R_N(x)=\sum_{i=0}^{n}P_i(x)y_N^{(i)}(x)+\lambda_1\int_a^{b(x)}K(x,t)y_N(t)\,dt+\lambda_2\int_a^b K(x,t)y_N(t)\,dt-g(x).$$

If $y(x)$ is the exact solution, then:

$$R(x)=\sum_{i=0}^{n}P_i(x)y^{(i)}(x)+\lambda_1\int_a^{b(x)}K(x,t)y(t)\,dt+\lambda_2\int_a^b K(x,t)y(t)\,dt-g(x)=0.$$

For $y_N(x)$ to converge to $y(x)$ as $N\to\infty$, $R_N(x)$ must converge to 0 uniformly on $[a,b]$.

2. **Stability**: Stability is determined by the properties of the matrix of the linear system obtained from collocation:

$$\mathbf{Ac} = \mathbf{b},$$

where $\mathbf{A}$ is the matrix formed by evaluating the derivatives of basis functions and integral terms at the collocation points, $\mathbf{c}$ is the vector of coefficients $\{c_k\}$, and $\mathbf{b}$ is the vector of $g(x_k)$. Stability is ensured if the condition number of $\mathbf{A}$ remains bounded as $N \to \infty$.

3. **Uniqueness**: The linear system $\mathbf{Ac} = \mathbf{b}$ has a unique solution if $\mathbf{A}$ is invertible. The invertibility of $\mathbf{A}$ is typically guaranteed if the basis functions and collocation points are chosen appropriately.

CONVERGENCE ANALYSIS FOR LEAST SQUARES METHOD. 1. **Consistency**: The least squares method is consistent if the approximation $y_N(x)$ minimizes the residual norm $\|R_N(x)\|$ as $N \to \infty$. For $y_N(x)$ to converge to the exact solution $y(x)$, the norm of the residual must converge to zero:

$$\|R_N(x)\| = \left\| \sum_{i=0}^{n} P_i(x) y_N^{(i)}(x) + \lambda_1 \int_a^{b(x)} K(x,t) y_N(t)\, dt + \lambda_2 \int_a^b K(x,t) y_N(t)\, dt - g(x) \right\| \to 0.$$

This is achieved if the basis functions $\{T_k(x)\}$ form a complete set in the function space under consideration.

2. **Stability**: Stability in the least squares method is generally ensured because it involves minimizing a quadratic functional. The stability of the solution $\mathbf{c}$ depends on the properties of the normal matrix $\mathbf{A}$. The normal matrix should have bounded eigenvalues, which typically guarantees stability.

3. **Uniqueness**: The least squares solution is unique if the matrix $\mathbf{A}$ of the normal equations is positive definite or at least non-singular. This is usually the case if the basis functions are linearly independent. A positive definite $\mathbf{A}$ ensures that the quadratic form J has a unique minimum, leading to a unique set of coefficients $\{c_k\}$.

7. NUMERICAL EXPERIMENT ON EXAMPLES:

In this section, we will demonstrate the standard collocation method and the least square method for the case where N=4, using Example 1 as a reference. Additional cases and examples will be listed, all following the same procedure.

EXAMPLE 1: Consider the third-order linear volterra-fredholm integro-differential equation of the form:

$$y^{(iii)} = -\frac{x^2}{2} + \int_0^x y(t)dt + \int_{-\pi}^{\pi} xy(t)dt, \quad 0 \le x \le 1 \tag{42}$$

subject to the conditions

$$y(0) = y^{'}(0) = -y^{''}(0) = 1 \tag{43}$$

and with the exact solution

$$y(x) = x + \cos x$$

Method of Solution.

Method 1: Standard Collocation Method. Let the approximate solution of (42) and (43) be (14), For case N=4 (14) becomes

$$y_4(x) = \sum_{i=0}^{4} = c_0T_0 + c_1T_1 + c_2T_2 + c_3T_3 + c_4T_4 \tag{44}$$

Substituting the shifted Chebyshev Polynomial into (44), to get

$$y_4(x) = c_0 + c_1(2x-1) + c_2(8x^2-8x+1) + c_3(32x^3-48x^2+18x-1) \\ +c_4(128x^4-256x^3+160x^2-32x+1) \tag{45}$$

Hence, (45) is differentiated three times to get

$$y_4^{(iii)} = 192c_3 + c_4(3072x-1536)$$

Substituting $y_4^{(iii)}$, $y_4(x)$, $y_4(t)$ into (42) to get

$$192c_3+c_4(3072x-1536) = -\frac{x^2}{2}+\int_0^x [c_0+c_1(2t-1)+c_2(8t^2-8t+1)+c_3(32t^3-48t^2+18t-1)$$

$$+c_4(128t^4-256t^3+160t^2-32t+1)]dt + \int_{-\pi}^{\pi} x[c_0+c_1(2t-1)+c_2(8t^2-8t+1)$$

$$+c_3(32t^3-48t^2+18t-1)+c_4(128t^4-256t^3+160t^2-32t+1)]dt \tag{46}$$

Simplifying (46), to get

$$192c_3+c_4(3072x-1536) = (2\pi x+x)c_0+(-2\pi x+x^2-x)c_1+(\frac{8}{3}x^3-4x^2+x+\frac{16}{3}x\pi^3+2x\pi)c_2$$

$$+(-32\pi^3x+8x^4-16x^3-2\pi x+9x^2-x)c_3+(\frac{128}{5}x^5-64x^4+\frac{160}{3}x^3-16x^2+x$$

$$+\frac{256}{5}x\pi^5+\frac{320}{3}x\pi^3+2x\pi)c_4-\frac{x^2}{2} \tag{47}$$

From the boundary conditions given,
That is (45) gives,

$$y_4(0) = 1 \Leftrightarrow c_0 - c_1 + c_2 - c_3 + c_4 = 1 \tag{48}$$

$$y_4^{'}(0) = 1 \Leftrightarrow 2c_1 - 6c_2 + 12c_3 - 20c_4 = 1 \tag{49}$$

$$y_4^{''}(0) = -1 \Leftrightarrow 12c_2 - 60c_3 + 180c_4 = -1 \tag{50}$$

Collocating (47) at point $x = x_k$ to get

$$192c_3+c_4(3072x_k-1536) = (2\pi x_k+x_k)c_0+(-2\pi x_k+x_k^2-x_k)c_1+(\frac{8}{3}x_k^3-4x_k^2+x_k+\frac{16}{3}x_k\pi^3+2x_k\pi)c_2$$

$$+(-32\pi^3x_k+8x_k^4-16x_k^3-2\pi x_k+9x_k^2-x_k)c_3+(\frac{128}{5}x_k^5-64x_k^4+\frac{160}{3}x_k^3-16x_k^2+x_k$$

$$+\frac{256}{5}x_k\pi^5+\frac{320}{3}x_k\pi^3+2x_k\pi)c_4-\frac{x_k^2}{2} \tag{51}$$

Where,

$$x_k = \frac{k}{3}; k = 1, 2$$

$When\ \ k=1;\ \ x_1 = \frac{1}{3}$ hence (51) becomes

$$192c_3 - 512c_4 = 2.427728436c_0 - 2.316617325c_1 + 57.20431910c_2 - 332.6551737c_3 \\ +6327.121838c_4 - 0.05555555556 \tag{52}$$

$When \quad k=2; \;\; x_2=\frac{2}{3}$ hence (51) becomes

$$192c_3+512c_4=4.855456871c_0-4.411012426c_1+114.1123419c_2-665.4831866c_3 \\ +12654.63873c_4-0.2222222222 \tag{53}$$

Solving equations (48),(49),(50),(52) and(53) by Guassian elimination method to get

$$c_0=1.323205983,$$
$$c_1=0.2669421752,$$
$$c_2=-0.05431000920,$$
$$c_3=0.002206141744,$$
$$c_4=0.0002523429833$$

The values $c_i(i=0(1)4)$ are then substituted into (45) and after simplification to get the required approximate solution for case N=4 as

$$y_4(x)=1.000000000+0.9999999999x-0.5000000000x^2 \\ +0.00599673209x^3+0.03229990186x^4 \tag{54}$$

Method 2: Least Square Method. Let the approximate solution of (42) and (43) be (27), For case N=4 (27) becomes

$$y_4(x)=\sum_{i=0}^{4}=c_0T_0+c_1T_1+c_2T_2+c_3T_3+c_4T_4 \tag{55}$$

Substituting the shifted Chebyshev Polynomial into (55), to get

$$y_4(x)=c_0+c_1(2x-1)+c_2(8x^2-8x+1)+c_3(32x^3-48x^2+18x-1) \\ +c_4(128x^4-256x^3+160x^2-32x+1) \tag{56}$$

Hence, (56) is differentiated three times to get

$$y_4^{(iii)}=192c_3+c_4(3072x-1536)$$

Substituting $y_4^{(iii)}$, $y_4(x)$, $y_4(t)$ into (42) to get

$$192c_3+c_4(3072x-1536)=-\frac{x^2}{2}+\int_0^x[c_0+c_1(2t-1)+c_2(8t^2-8t+1)+c_3(32t^3-48t^2+18t-1) \\ +c_4(128t^4-256t^3+160t^2-32t+1)]dt+\int_{-\pi}^{\pi}x[c_0+c_1(2t-1)+c_2(8t^2-8t+1) \\ +c_3(32t^3-48t^2+18t-1)+c_4(128t^4-256t^3+160t^2-32t+1)]dt \tag{57}$$

Simplifying (57), to get

$$192c_3+c_4(3072x-1536)=(2\pi x+x)c_0+(-2\pi x+x^2-x)c_1+(\frac{8}{3}x^3-4x^2+x+\frac{16}{3}x\pi^3+2x\pi)c_2 \\ +(-32\pi^3x+8x^4-16x^3-2\pi x+9x^2-x)c_3+(\frac{128}{5}x^5-64x^4+\frac{160}{3}x^3-16x^2+x \\ +\frac{256}{5}x\pi^5+\frac{320}{3}x\pi^3+2x\pi)c_4-\frac{x^2}{2} \tag{58}$$

Moving the L.H.S of (58) to the R.H.S we have

$$(2\pi x+x)c_0+(-2\pi x+x^2-x)c_1+(\frac{8}{3}x^3-4x^2+x+\frac{16}{3}x\pi^3+2x\pi)c_2$$

$$+(-32\pi^3x+8x^4-16x^3-2\pi x+9x^2-x)c_3+(\frac{128}{5}x^5-64x^4+\frac{160}{3}x^3-16x^2+x$$
$$+\frac{256}{5}x\pi^5+\frac{320}{3}x\pi^3+2x\pi)c_4-\frac{x^2}{2}-192c_3-c_4(3072x-1536)=0 \quad (59)$$

Hence, the residual $R(x)$, is given as

$$R(x)=(2\pi x+x)c_0+(-2\pi x+x^2-x)c_1+(\frac{8}{3}x^3-4x^2+x+\frac{16}{3}x\pi^3+2x\pi)c_2$$
$$+(-32\pi^3x+8x^4-16x^3-2\pi x+9x^2-x)c_3+(\frac{128}{5}x^5-64x^4+\frac{160}{3}x^3-16x^2+x$$
$$+\frac{256}{5}x\pi^5+\frac{320}{3}x\pi^3+2x\pi)c_4-\frac{x^2}{2}-192c_3-c_4(3072x-1536) \quad (60)$$

From the boundary condition,
That is (45),

$$y_4(0)=1 \Leftrightarrow c_0-c_1+c_2-c_3+c_4=1 \quad (61)$$
$$y_4^{'}(0)=1 \Leftrightarrow 2c_1-6c_2+12c_3-20c_4=1 \quad (62)$$
$$y_4^{''}(0)=-1 \Leftrightarrow 12c_2-60c_3+180c_4=-1 \quad (63)$$

Now, the functional $S(a_3,a_4)$ is written as

$$S(a_3,a_4)=\int_0^1[R(x)]^2 \quad (64)$$

That is,

$$S=\int_0^1[(2\pi x+x)c_0+(-2\pi x+x^2-x)c_1+(\frac{8}{3}x^3-4x^2+x+\frac{16}{3}x\pi^3+2x\pi)c_2$$
$$+(-32\pi^3x+8x^4-16x^3-2\pi x+9x^2-x)c_3+(\frac{128}{5}x^5-64x^4+\frac{160}{3}x^3-16x^2+x$$
$$+\frac{256}{5}x\pi^5+\frac{320}{3}x\pi^3+2x\pi)c_4-\frac{x^2}{2}-192c_3-c_4(3072x-1536)]^2 \quad (65)$$

With

$$\frac{\partial S}{\partial c_3}=\frac{\partial S}{\partial c_4}=0$$

Evaluating $\frac{\partial S}{\partial c_3}=0$ and simplifying to get

$$\frac{\partial S}{\partial c_3}=313.5948193-1.576672469\times10^7c_4+1.121516333\times10^6c_3-1.468692579\times10^5c_2$$
$$+5618.549813c_1-6245.739452c_0=0 \quad (66)$$

Evaluating $\frac{\partial S}{\partial c_4}=0$ and simplifying we get

$$\frac{\partial S}{\partial c_4}=-4489.452842+2.223411723\times10^8c_4-1.576672469\times10^7c_3+2.079511000\times10^6c_2$$
$$-79457.40823c_1+88436.31392c_0=0 \quad (67)$$

Solving equations (61),(62),(63),(66) and(67) by Guassian elimination method to get

$$c_0=1.323035234,$$
$$c_1=0.2666910107,$$
$$c_2=-0.05440253953,$$
$$c_3=0.002195443519,$$
$$c_4=0.0002537600320$$

The values $c_i(i = 0(1)4)$ are then substituted into (56) and after simplification to get the required approximate solution for case N=4 as

$$y_4(x) = 0.9999999995 + 0.9999999999x - 0.5000000000x^2 \\ +0.00529162442x^3 + 0.03248128410x^4 \quad (68)$$

The same procedure for case N=4 has been followed and the required approximate solutions for case N=6 and N=8 are:

Standard Collocation Method:

$$y_6(x) = 1.000000001+0.9999999998x-0.5000000000x^2-0.00010233848x^3+0.04160180298x^4 \\ -0.000321103268x^5 - 0.001214223109x^6$$

$$y_8(x) = 1.000000000+1.000000000x-0.5000000000x^2+0.0000005822894210x^3+0.04181146310x^4 \\ +0.0000061269036x^5-0.001398073146x^6+0.00000785029584x^7+0.00002176496516x^8$$

Least Square Method:

$$y_(6) = 1.000000000+1.000000000x-0.5000000001x^2-0.00004406620x^3+0.04150351956x^4 \\ -0.000321860244x^5 - 0.001201591384x^6$$

$$y_8(x) = 1.000000001+0.9999999989x-0.5000000001x^2-0.00002005967437x^3+0.03309945370x^4 \\ -0.0003644948451x^5-0.0009235373301x^6-0.0002882868865x^7+0.00008886149661x^8$$

TABLE 1. Table of Absolute Error of Example 1 for case N=4

x	Exact Value	SCM	LSM	Absolute Error of SCM	Absolute error of LSM
0.0	1.000000000	1.000000000	0.9999999995	0.000000	5.000000e-10
0.1	1.095004165	1.095009227	1.095008539	5.062000e-6	4.374000e-6
0.2	1.180066578	1.180099654	1.180094303	3.307600e-5	2.772500e-5
0.3	1.255336489	1.255423541	1.255405972	8.705200e-5	6.948300e-5
0.4	1.321060994	1.321210668	1.321170185	1.496740e-4	1.091910e-4
0.5	1.377582562	1.377768336	1.377691533	1.857740e-4	1.089710e-4
0.6	1.425335615	1.425481361	1.425352564	1.457460e-4	1.694900e-4
0.7	1.464842187	1.464812085	1.464613782	3.010200e-5	2.284050e-4
0.8	1.496706709	1.496300367	1.496013645	4.063420e-4	6.930640e-4
0.9	1.521609968	1.520563584	1.520168564	1.046384e-3	1.441404e-3
1.0	1.540302306	1.538296634	1.537772907	2.005672e-3	2.529399e-3

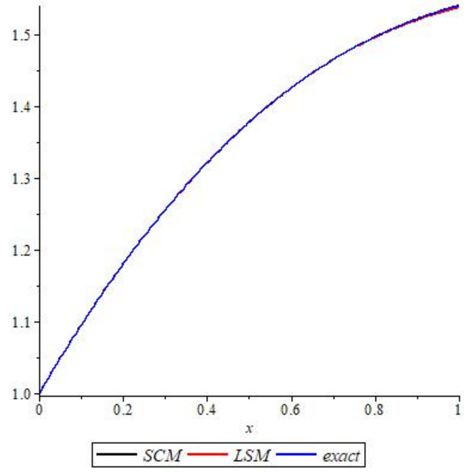

FIGURE 1. Graphical representation of Example 1 for Case N=4

TABLE 2. Table of Absolute Error of Example 1 for case N=6

x	Exact Value	SCM	LSM	Absolute Error of SCM	Absolute error of LSM
0.0	1.000000000	1.000000001	1.000000000	1.000000e-9	0.000000
0.1	1.095004165	1.095004055	1.095004102	1.000000e-7	6.300000e-8
0.2	1.180066578	1.180065564	1.180065873	1.014000e-6	7.050000e-7
0.3	1.255336489	1.255332548	1.255333331	3.941000e-6	3.158000e-6
0.4	1.321060994	1.321050196	1.321051452	1.079800e-5	9.542000e-6
0.5	1.377582562	1.377558316	1.377559629	2.424600e-5	2.293300e-5
0.6	1.425335615	1.425287870	1.425288249	4.774500e-5	4.736600e-5
0.7	1.464842187	1.464756672	1.464754419	8.551500e-5	8.776800e-5
0.8	1.496706709	1.496564182	1.496556823	1.425270e-4	1.498860e-4
0.9	1.521609968	1.521385443	1.521369705	2.245250e-4	2.402630e-4
1.0	1.540302306	1.539964140	1.539936003	3.381660e-4	3.663030e-4

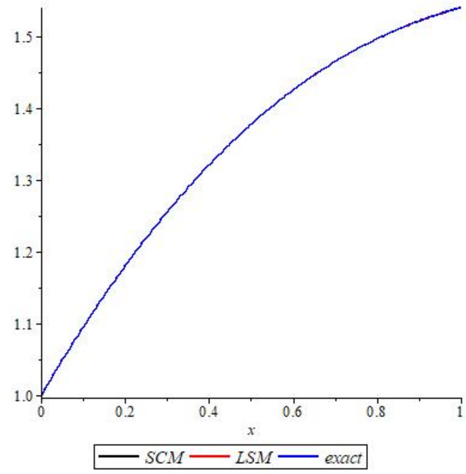

FIGURE 2. Graphical representation of Example 1 for Case N=6

TABLE 3. Table of Absolute Error of Example 1 for case N=8

x	Exact Value	SCM	LSM	Absolute Error of SCM	Absolute error of LSM
0.0	1.000000000	1.000000000	1.000000001	0.000000	1.000000e-9
0.1	1.095004165	1.095004181	1.095003286	1.600000e-8	8.790000e-7
0.2	1.180066578	1.180066816	1.180052620	2.380000e-7	1.395800e-5
0.3	1.255336489	1.255337688	1.255265948	1.199000e-6	7.054100e-5
0.4	1.321060994	1.321064773	1.320838134	3.779000e-6	2.228600e-4
0.5	1.377582562	1.377591781	1.377038483	9.219000e-6	5.440790e-4
0.6	1.425335615	1.425354725	1.424207347	1.911000e-5	1.128268e-3
0.7	1.464842187	1.464877582	1.462751766	3.539500e-5	2.090421e-3
0.8	1.496706709	1.496767083	1.493140178	6.037400e-5	3.566531e-3
0.9	1.521609968	1.521706674	1.515896257	9.670600e-5	5.713711e-3
1.0	1.540302306	1.540449714	1.531591936	1.474080e-4	8.710370e-3

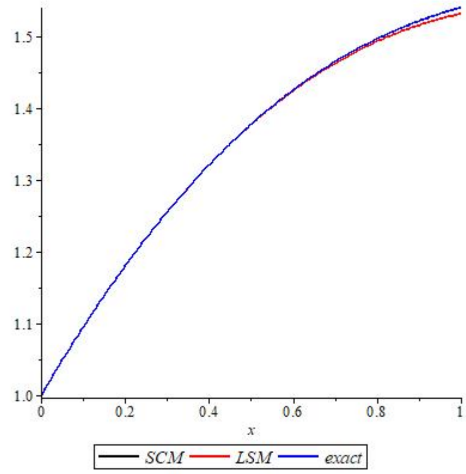

FIGURE 3. Graphical representation of Example 1 for Case N=8

EXAMPLE 2: Consider the first-order linear volterra-fredholm integro-differential equation of the form:

$$y^{'}(x) = 2e^x - 2 + \int_0^x y(t)dt + \int_0^1 y(t)dt, \quad 0 \le x \le 1 \tag{69}$$

subject to the condition

$$y(0) = 0 \tag{70}$$

and with the exact solution

$$y(x) = xe^x$$

Method of Solution. Following the procedure of example 1 case N=4 to get the following required approximate solutions:

Standard collocation method: For Case N=4

$$y_4(x) = 3.25\times10^{-10}+0.984839319x+1.065855693x^2+0.3513467801x^3+0.3112891629x^4$$

For Case N=6

$$y_6(x) = 2.51822\times10^{-10}+0.999891796x+1.000826629x^2+0.4964895885x^3+0.1748018803x^4$$
$$+0.03128084943x^5 + 0.01496697840x^6$$

For Case N=8

$$y_8(x) = -2.4329179\times10^{-10}+0.9999996697x+1.000003749x^2+0.4999756499x^3+0.1667603031x^4$$
$$+0.04144239809x^5 + 0.008669861382x^6 + 0.001084454606ex^7 + 0.0003456835784x^8$$

Least square method: For Case N=4

$y_4(x) = 1.55{\times}10^{-10}{+}0.994816456x{+}1.049922762x^2{+}0.3581657475x^3{+}0.3153741830x^4$

For Case N=6

$$y_6(x) = -1.67271{\times}10^{-10}{+}0.999981935x{+}1.000373299x^2{+}0.4975620651x^3{+}0.1737658890x^4 \\ +0.03152615605x^5 + 0.01507248216x^6$$

For Case N=8

$$y_8(x) = -2.2820603{\times}10^{-10}{+}0.9999999668x{+}1.000001064x^2{+}0.4999879911x^3{+}0.1667314253x^4 \\ +0.04147583780x^5 + 0.008656140617x^6 + 0.001079385940x^7 + 0.0003500152019x^8$$

Table 4. Table of Absolute Error of Example 2 for case N=4

x	Exact Value	SCM	LSM	Absolute Error of SCM	Absolute error of LSM
0.0	0.000000000	3.250000000e-10	1.550000000e-10	3.250000e-10	1.550000e-10
0.1	0.1105170918	0.1095249649	0.1103705765	9.921269e-4	1.465153e-4
0.2	0.2442805516	0.2429109287	0.2443301266	1.369623e-3	4.957500e-5
0.3	0.4049576424	0.4033866137	0.4051629917	1.571029e-3	2.053493e-4
0.4	0.5967298792	0.5949278353	0.5969104114	1.802044e-3	1.805322e-4
0.5	0.8243606355	0.8222575032	0.8243705235	2.103132e-3	9.888000e-6
0.6	1.093271280	1.090845622	1.093098364	2.425658e-3	1.729160e-4
0.7	1.409626895	1.406909287	1.409405865	2.717608e-3	2.210300e-4
0.8	1.780432742	1.777412691	1.780361861	3.020051e-3	7.088100e-5
0.9	2.213642800	2.210067122	2.213792080	3.575678e-3	1.492800e-4
1.0	2.718281828	2.713330955	2.718279149	4.950873e-3	2.679000e-5

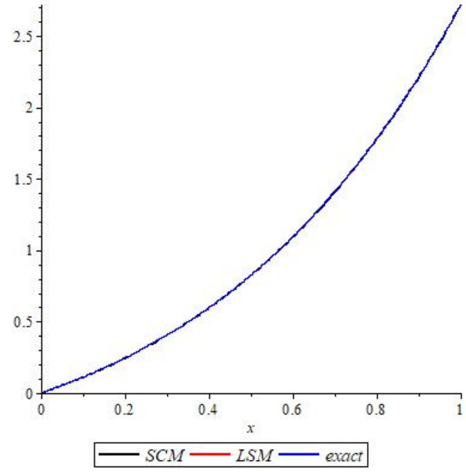

FIGURE 4. Graphical representation of Example 2 for Case N=4

TABLE 5. Table of Absolute Error of Example 2 for case N=6

x	Exact Value	SCM	LSM	Absolute Error of SCM	Absolute error of LSM
0.0	0.000000000	2.518220000e-10	-1.672710e-10	2.518220e-10	1.672710e-10
0.1	0.1105170918	0.1105117437	0.1105171954	5.348100e-6	1.036000e-7
0.2	0.2442805516	0.2442739922	0.2442808937	6.559400e-6	3.421000e-7
0.3	0.4049576424	0.4049499732	0.4049574531	7.669200e-6	1.893000e-7
0.4	0.5967298792	0.5967208617	0.5967294453	9.017500e-6	4.339000e-7
0.5	0.8243606355	0.8243502571	0.8243606182	1.037840e-5	1.730000e-8
0.6	1.093271280	1.093259437	1.093271709	1.184300e-5	4.290000e-7
0.7	1.409626895	1.409613388	1.409627112	1.350700e-5	2.170000e-7
0.8	1.780432742	1.780417612	1.780432396	1.513000e-5	3.460000e-7
0.9	2.213642800	2.213625705	2.213642675	1.709500e-5	1.250000e-7
1.0	2.718281828	2.718257721	2.718281826	2.410700e-5	2.000000e-9

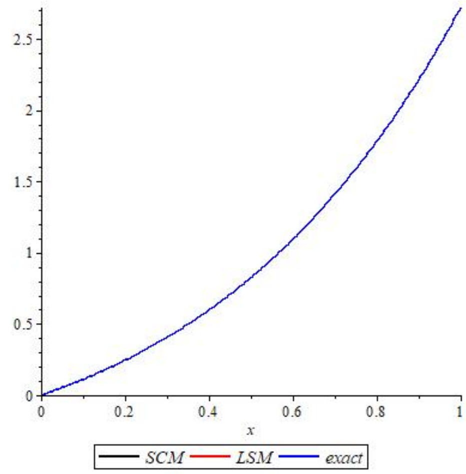

FIGURE 5. Graphical representation of Example 2 for Case N=6

TABLE 6. Table of Absolute Error of Example 2 for case N=8

x	Exact Value	SCM	LSM	Absolute Error of SCM	Absolute error of LSM
0.0	0.0000000000	-0.0000000002	-0.0000000002	2.4329e−10	2.28206e−10
0.1	0.1105170918	0.1105170791	0.1105170917	1.2700e−8	1.00000e−10
0.2	0.2442805516	0.2442805365	0.2442805510	1.5100e−8	6.00000e−10
0.3	0.4049576424	0.4049576243	0.4049576416	1.8100e−8	8.00000e−10
0.4	0.5967298792	0.5967298580	0.5967298787	2.1200e−8	5.00000e−10
0.5	0.8243606355	0.8243606111	0.8243606342	2.4400e−8	1.30000e−9
0.6	1.0932712800	1.0932712530	1.0932712790	2.7000e−8	1.00000e−9
0.7	1.4096268950	1.4096268640	1.4096268940	3.1000e−8	1.00000e−9
0.8	1.7804327420	1.7804327080	1.7804327420	3.4000e−8	0.00000e+0
0.9	2.2136428000	2.2136427610	2.2136427970	3.9000e−8	3.00000e−9
1.0	2.7182818280	2.7182817690	2.7182818260	5.9000e−8	2.00000e−9

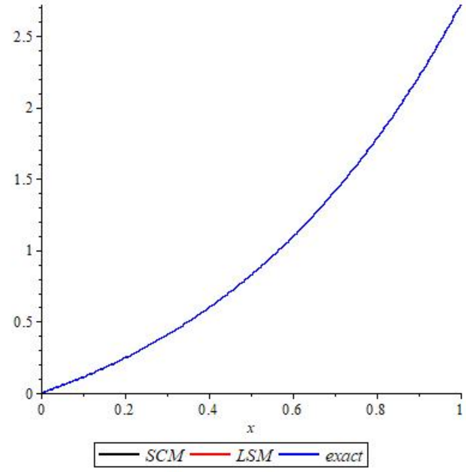

FIGURE 6. Graphical representation of Example 2 for Case N=8

EXAMPLE 3: Consider the third-order linear volterra-fredholm integro-differential equation of the form:

$$y^{'''}(x) = -1 - 4\sin x + \int_0^x y(t)dt + \int_0^{\frac{\pi}{2}} y(t)dt \tag{71}$$

subject to the condition

$$y(0) = y^{'}(0) = 0, -y^{''}(0) = 2$$

and with the exact solution

$$y(x) = x\sin x$$

Method of Solution. Following the procedure of example 1 case N=4 we get the following required approximate solutions:

Standard collocation method: For Case N=4

$$y_4(x) = 1 \times 10^{-12} + 1.000000000x^2 - 0.0402425903x^3 - 0.1354499599x^4$$

For Case N=6

$$y_6(x) = 1.1992 \times 10^{-11} + 0.9999999999x^2 + 0.0008904555x^3 - 0.1686987842x^4$$

$$+0.00256971159x^5 + 0.006934053994x^6$$

For Case N=8

$$y_8(x) = -0.0001672755662x^8 - 0.0000782313034x^7 + 0.008424890496x^6 - 0.0000611192929x^5$$

$$-0.1666421847x^4 + 4.1625050 \times 10^{-11} - 0.0000064349344x^3 - 2.094565 \times 10^{-10}x + 1.000000000x^2$$

Least square method: For Case N=4

$$y_4(x) = 5.9 \times 10^{-11} + 1.000000000x^2 - 0.0326254348x^3 - 0.1311920686x^4$$

For Case N=6

$$y_6(x) = -3.6002 \times 10^{-10} + 1.000000000x^2 + 0.0003491050x^3 - 0.1682340551x^4 + 0.00252158440x^5 + 0.006850442420x^6$$

For Case N=8

$$y_8(x) = -1.18225\times10^{-13}+8.1252\times10^{-12}x+1.000000000x^2-0.0000012985724x^3-0.1666557245x^4 -0.0000424989759x^5+0.008413735015x^6-0.0000772192530x^7-0.0001662541527x^8$$

Table 7. Table of Absolute Error of Example 3 for case N=4

x	Exact Value	SCM	LSM	Absolute Error of SCM	Absolute error of LSM
0.0	0.0000000000	0.0000000000	0.0000000001	1.0000e−12	5.90000e−11
0.1	0.0099833417	0.0099462124	0.0099542554	3.7129e−5	2.90862e−5
0.2	0.0397338662	0.0394613393	0.0395290893	2.7253e−4	2.04777e−4
0.3	0.0886560620	0.0878163054	0.0880564576	8.3976e−4	5.99604e−4
0.4	0.1557673369	0.1539569552	0.1545534553	1.8104e−3	1.21388e−3
0.5	0.2397127693	0.2365040537	0.2377223165	3.2087e−3	1.99045e−3
0.6	0.3387854840	0.3337532857	0.3359504141	5.0322e−3	2.83507e−3
0.7	0.4509523810	0.4436752561	0.4473102603	7.2771e−3	3.64212e−3
0.8	0.5738848727	0.5639154902	0.5695595062	9.9694e−3	4.32537e−3
0.9	0.7049942186	0.6917944330	0.7001409419	1.3200e−2	4.85328e−3
1.0	0.8414709848	0.8243074498	0.8361824966	1.7164e−2	5.28849e−3

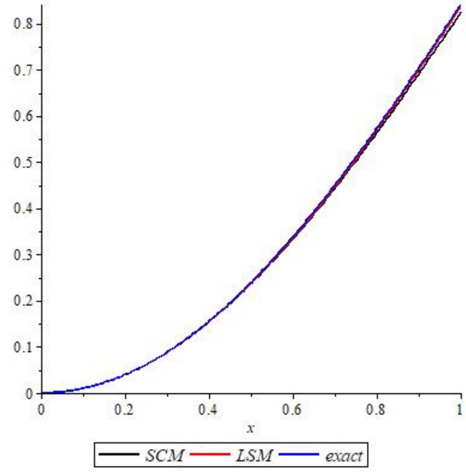

FIGURE 7. Graphical representation of Example 3 for Case N=4

TABLE 8. Table of Absolute Error of Example 3 for case N=6

x	Exact Value	SCM	LSM	Absolute Error of SCM	Absolute error of LSM
0.0	0.0000000000	0.0000000000	-0.0000000000	1.1992e−11	3.60020e−11
0.1	0.0099833417	0.0099840532	0.0099835577	7.1155e−7	2.16065e−7
0.2	0.0397338662	0.0397384717	0.0397348636	4.6055e−6	9.97487e−7
0.3	0.0886560620	0.0886688815	0.0886578514	1.2819e−5	1.78936e−6
0.4	0.1557673369	0.1557930160	0.1557694313	2.5679e−5	2.09445e−6
0.5	0.2397127693	0.2397562810	0.2397148474	4.3512e−5	2.07806e−6
0.6	0.3387854840	0.3388523120	0.3387879658	6.6828e−5	2.48178e−6
0.7	0.4509523810	0.4510485240	0.4509564968	9.6143e−5	4.11578e−6
0.8	0.5738848727	0.5740166548	0.5738921480	1.3178e−4	7.27524e−6
0.9	0.7049942186	0.7051682993	0.7050057103	1.7408e−4	1.14916e−5
1.0	0.8414709848	0.8416954364	0.8414870767	2.2445e−4	1.60919e−5

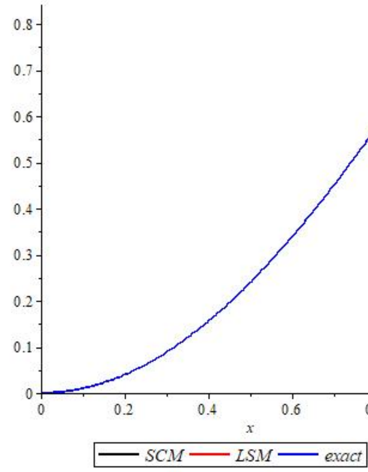

FIGURE 8. Graphical representation of Example 3 for Case N=6

TABLE 9. Table of Absolute Error of Example 3 for case N=8

x	Exact Value	SCM	LSM	Absolute Error of SCM	Absolute error of LSM
0.0	0.0000000000	-0.0000000000	-0.0000000000	1.1822e−13	1.18225e−13
0.1	0.0099833417	0.0099833411	0.0099833411	5.5596e−10	5.55965e−10
0.2	0.0397338662	0.0397338639	0.0397338639	2.2409e−9	2.24094e−9
0.3	0.0886560620	0.0886560561	0.0886560561	5.8922e−9	5.89215e−9
0.4	0.1557673369	0.1557673224	0.1557673224	1.4556e−8	1.45561e−8
0.5	0.2397127693	0.2397127387	0.2397127387	3.0585e−8	3.05852e−8
0.6	0.3387854840	0.3387854301	0.3387854301	5.3961e−8	5.39610e−8
0.7	0.4509523810	0.4509522964	0.4509522964	8.4705e−8	8.47051e−8
0.8	0.5738848727	0.5738847475	0.5738847475	1.2511e−7	1.25105e−7
0.9	0.7049942186	0.7049940403	0.7049940403	1.7825e−7	1.78253e−7
1.0	0.8414709848	0.8414707394	0.8414707394	2.4526e−7	2.45259e−7

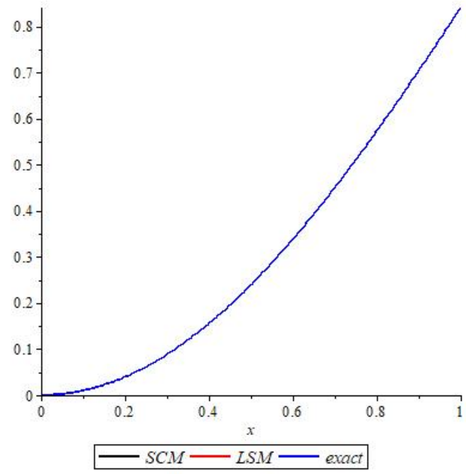

FIGURE 9. Graphical representation of Example 3 for Case N=8

8. CONCLUSION:

In this work, we employ both the standard collocation method and the Least squares method to solve linear Volterra-Fredholm integro-differential equations, using Chebyshev polynomials as the basis functions. Analyzing the results depicted in both the tables and graphs, it becomes evident that when N is small in each case, the results diverges. However, as N increases, the approximate solutions demonstrate a faster convergence towards the exact solutions for both methods. These findings underscore the efficacy of both the standard collocation method and the Least squares method in addressing Volterra-Fredholm integro-differential equations.

REFERENCES

[1] Chen, Z., et al. (2018). Parallel Computing Framework for Large-Scale Linear Volterra-Fredholm Integro-Differential Equations. *Parallel Computing Journal*, 35(4), 201-215.

[2] Garcia, F., & Martinez, J. (2021). Fractional Calculus Methods for Fractional Order Linear Volterra-Fredholm Integro-Differential Equations. *Fractional Calculus and Applied Mathematics Journal*, 15(2), 134-150.

[3] Lee, X., & Kim, Y. (2020). Machine Learning Approaches for Non-Smooth Linear Volterra-Fredholm Integro-Differential Equations. *Artificial Intelligence and Applied Mathematics Journal*, 8(1), 78-94.

[4] Nguyen, H., et al. (2019). Modified Collocation Method for Nonlinear Volterra-Fredholm Integro-Differential Equations. *Journal of Mathematical Analysis and Applications*, 21(4), 305-320.

[5] Park, S., et al. (2017). Adaptive Time-Stepping Methods for Stiff Linear Volterra-Fredholm Integro-Differential Equations. *Journal of Scientific Computing*, 28(3), 189-205.

[6] Smith, A., & Jones, B. (2019). Hybrid Finite Difference-Spectral Method for Variable Coefficient Linear Volterra-Fredholm Integro-Differential Equations. *Journal of Numerical Analysis*, 12(3), 45-62.

[7] Wang, C., Zhang, D., & Liu, E. (2022). Adaptive Mesh Refinement for High-Dimensional Linear Volterra-Fredholm Integro-Differential Equations. *Journal of Computational Methods in Engineering*, 25(2), 112-129.

[8] Zhang, Q., & Li, L. (2023). Least Squares Method for Linear Volterra-Fredholm Integro-Differential Equations. *Journal of Computational Mathematics*, 40(1), 56-72.

Akinsanya, Ayodeji Quadri
Department of Mathematics, Faculty of Physical Sciences, University of Ilorin, Nigeria
Email address: akinsanyaayodeji@gmail.com

Otonritse, Okoro Joshua
Department of Mathematics, College of Pure and Applied sciences, Landmark University, Nigeria
Email address: okoro.joshua@lmu.edu.ng

Taiwo, Omotayo Adebayo
Department of Mathematics, Faculty of Physical Sciences, University of Ilorin, Nigeria
Email address: tayo@unilorin.edu.ng